1972-09

A comparison of classical and Bayesian methods for determining lower confidence limits on system reliability.

Kirk, Gary Lee

A COMPARISON OF CLASSICAL AND BAYESIAN METHODS FOR DETERMINING LOWER CONFIDENCE LIMITS ON SYSTEM RELIABILITY

Gary Lee Kirk

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A COMPARISON OF CLASSICAL AND BAYESIAN
METHODS FOR DETERMINING LOWER
CONFIDENCE LIMITS ON SYSTEM RELIABILITY

by

Gary Lee Kirk

Thesis Advisor: W. M. Woods

September 1972

A Comparison of Classical and Bayesian Methods for Determining Lower Confidence Limits on System Reliability

by

Gary Lee Kirk
Lieutenant Commander, United States Navy
B.A., University of Kansas, 1963

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1972

ABSTRACT

A series system is simulated to obtain lower confidence limits on system reliability using Bayesian techniques. A comparison between classical and Bayesian methods is made. Random beta variate generators are developed and used in the simulation. The results of the simulation are tabulated for easy comparison of the Bayesian and classical methods. The values of lower confidence limits that are realized using the Bayesian method decrease as the number of components increase. In most cases, as the number of components increase, the Bayesian method appears to yield lower values of lower confidence limits than the classical method.

TABLE OF CONTENTS

I. INTRODUCTION

In a series system the reliability of the system, RS, depends upon the reliability of each of the k components or subsystems of which it is composed. This can be modeled as follows:

$$RS = \prod_{i=1}^{k} R_i ,$$

where R_i is the reliability of the ith component. Using success/fail test data classical or Bayesian methods can be used to compute lower confidence limits on RS. In a simple system with few components the Bayesian method offers many advantages over the more conservative classical method. However, as the system becomes more complex with a higher number of components, problems start to arise with the use of standard Bayesian techniques.

The purpose of this paper is to compare classical and Bayesian methods of computing lower confidence limits. Computer simulations were used to determine lower confidence limits on system reliability using Bayesian methods. A Poisson approximation was used to obtain lower confidence limits for the classical approach. Several factors were varied but the main interest of the investigation was the effect of the number of components of a system on the values of lower confidence limits on system reliability.

II. METHODS

In order to compute lower confidence limits on system reliability of a series system two methods can be used. The first method to be discussed is the classical approach which is based solely on test results. The second method is the Bayesian approach which utilizes assumptions based on prior knowledge of similar systems.

A. CLASSICAL METHOD

The classical approach to system reliability uses only the mission test data on each component. These results are then combined in order to obtain an estimate of the system reliability. In the most simple case where k components are each tested n times and there are no failures in any of the components it is assumed that this is equivalent to testing the entire system n times with no failures. In this case the lower $100(1-\alpha)\%$ confidence limit can be found in the same manner as with just one item. This is done by solving for p in the equation:

$$\sum_{j=s}^{n} \binom{n}{j} p^{j} (1-p)^{n-j} = \alpha$$

where s = number of successes.

The solution is a lower $100(1-\alpha)\%$ confidence limit on RS.

In the case of zero failures $RS_{L(\alpha)} = \sqrt[N]{\alpha}$. When only one failure occurs the same procedure can be used. However,

when more than one failure occurs the procedure becomes more complicated.

In the case where each component is tested n times and there are few component failures, an approximation to the classical value of $RS_{L(\alpha)}$ can be found as follows.

Let $\quad Q_i = 1-R_i$,

then $\quad RS = \sum_{i=1}^{k} (1-Q_i)$.

If the Q_i's are small then

$$RS \doteq 1 - \sum_{i=1}^{k} Q_i .$$

Let f_i = the number of failures on the ith component. Since f_i is a sum of n Bernoulli trials it is binomial (n,Q_i). If Q_i is small then each f_i is approximately Poisson (nQ_i).

Let $F = \sum_{i=1}^{k} f_i$;

then F is approximately Poisson $(n \sum_{i=1}^{k} Q_i)$,

and the upper $100(1-\alpha)\%$ confidence limit [Ref. 1] for $\sum_{i=1}^{k} Q_i$ is $\dfrac{\chi^2_{\alpha}, 2(F+1)}{2n}$, where χ^2_{α}, 2(F+L) is the $100(1-\alpha)$th percentile of the χ^2 distribution with 2(F+1) degrees of freedom.

Thus $RS_{L(\alpha)} \doteq 1 - \frac{\chi^2_{\alpha}, 2(F+1)}{2n}$

is an approximation to the classical value for a lower $100(1-\alpha)\%$ confidence limit on RS.

B. BAYESIAN METHOD

The Bayesian approach to system reliability is to treat each component reliability, R_i, as a random variable. One particular method is to assume that each component reliability has a beta distribution. The beta density function can be defined as follows:

$$F(r;a,b) = \frac{1}{B(a,b)} r^{a-1}(1-r)^{b-1} \quad \text{for } 0 \le r \le 1,\ a > 0,\ b > 0,$$

where $$B(a,b) = \frac{\Gamma(a)\ \Gamma(b)}{\Gamma(a+b)} .$$

A prior beta density B(r;a,b) is assumed for each component before testing begins. The test data is then used to obtain a posterior beta density B(r;a+s,b+f) where s is the number of successful tests and f is the number of failures for the component.

By generating a random beta variate for each component based on its posterior density a value of RS is calculated using the model:

$$RS = \sum_{i=1}^{k} R_i .$$

A distribution of RS values is simulated by repeating this procedure. The lower confidence limits are thus estimated

by the appropriate percentile points of the simulated distributions of RS values.

The main advantage of the Bayesian method is that past experience with technically similar hardware can be used to determine appropriate priors for each component. This allows higher reliability goals to be met with less testing. However, some precautions must be observed in selecting the beta priors. Choosing a beta prior of B(r;a,b), where a and b are integers, is equivalent in the classical sense to assuming that (a+b-1) tests have been performed and that a of these were successes. It is equivalent because the αth percentile point of the B(r;a,b) distribution is the lower $100(1-\alpha)\%$ confidence limit for p when there are (a+b-1) Bernoulli trials, each with probability p of success, and a of these trials are successful. This is readily apparent from an identity that relates the beta distribution to the binomial distribution [Ref. 2].

As the sum of the beta parameters (a+b) in the prior density increases relative to the number of tests to be observed the resulting lower confidence limits are more of a function of the assumed prior density than of the test results. One procedure is to limit the sum of the two parameters to a function of the number of tests. If the b parameter is less than one a different problem exists even though the sum of the parameters is small. This problem can be shown by the following discussion.

For any α, a family of beta distributions exist such that the αth percentile point of each is the same. If the b parameter is set equal to 1, then for any $B(r;a,b)$ distribution an a^* can be found such that the αth percentile point of $B(r;a,b)$ and $B(r;a^*,1)$ are the same. The $B(r;a^*,1)$ distribution is convenient since it can be related to the classical case of having already tested a^* items with no failures. The problem when the b parameter is less than one can be shown by converting the initial $B(r;a,b)$ prior to a $B(r;a^*,1)$ prior. In this case if there are no failures the posterior distribution will be $B(r;a^*+s,1)$ where s is the number of successful tests. If the initial beta prior is not converted the posterior distribution will be $B(r;a+s,b)$, which can be converted to $B(r;(a+s)^*,1)$. In all cases that were investigated the $(a+s)^*$ was larger than a^*+s when the b parameter was less than one. Thus, in an equivalent classical sense, it appears that each successful test will be counted as more than one success. This amplification effect seems to increase rapidly as the b parameter approaches zero. Also if the beta prior ($B(r;a,b)$, where b is less than one, is converted to a beta $B(r;a^*,1)$ to ensure that each success counts only as one success, the resulting a^* may be too large. The details of the conversion are given in the next section and a table of the resulting a^*'s for various values of b can be found in Appendix A.

III. SIMULATION

The purpose of this investigation was to compare lower confidence limits derived by using Bayesian methods with those obtained by using the classical method. In order to simplify the simulation and to be able to make comparisons it was assumed that each of the components had the same beta prior density and that each component was mission tested the same number of times. Using the same beta prior for each component is not a necessary condition for the simulation to work.

A. SIMULATION PROCEDURE

A series system of k components was simulated in the following manner. Let $B1_i(r;a_i,b_i)$ be the initial beta prior for R_i. For this investigation the following three beta priors were used: $B(r;5.0,0.03)$, $B(r;5.0,0.05)$ and $B(r;5.0,0.10)$. For each $B1_i(r;a_i,b_i)$ a new beta prior, $B2_i(r;a_i^*,1)$, was computed such that the 10th percentile point of $B2_i(r;a_i^*,1)$ was the same as the 10th percentile point of $B1_i(r;a_i,b_i)$. The value for a_i^* is easily computed since if the b parameter is 1 then the percentile point (P) of a $B(x;a,1)$ distribution is given by the equation $P = x^a$. Thus $a_i^* = \ln(0.10)/\ln x$, where x is the 10th percentile point of $B1_i(r;a_i,b_i)$.

Let n_i = the number of trials for each component,

and s_i = the number of successes for each component.

The $B2_i(r;a_i^*,1)$ prior was then adjusted in the following manner:

$$B21_i(r;a_i,1) \quad \text{where } a_i = \min(0.75n_i,a_i^*)$$

$$B22_i(r;a_i,1) \quad \text{where } a_i = \min(1.0\ n_i,a_i^*)$$

$$B23_i(r;a_i,1) \quad \text{where } a_i = \min(1.5\ n_i,a_i^*)\ .$$

A beta posterior $B3_i(r;a_i,b_i)$ for each R_i was computed in four cases as follows:

Case 1 $B3_i(r;a_i,b_i) = B1_i(r;a_i+s_i,b_i+n_i-s_i)$

Case 2 $B3_i(r;a_i,b_i) = B21_i(r;a_i+s_i,1+n_i-s_i)$

Case 3 $B3_i(r;a_i,b_i) = B22_i(r;a_i+s_i,1+n_i-s_i)$

Case 4 $B3_i(r;a_i,b_i) = B23_i(r;a_i+s_i,1+n_i-s_i)$

By generating $B3_i(r;a_i,b_i)$ random variates, which represent the posterior distributions of each of the components, a value for RS was obtained by taking their product. This procedure was repeated 500 times and realized values for RS were then ordered. The lower $100(1-\alpha)\%$ confidence limits were then determined by selecting the appropriate percentile points of the simulated distribution on RS.

B. GENERATION OF RANDOM BETA VARIATES

Two means of generating random beta variates were used in the simulation. In the case where the b parameter of the $B(r;a,b)$ distribution was an integer an exponential generator was used to realize random beta variates. In the case

of noninteger parameters, Monte Carlo rejection techniques were used.

In order to generate random B(r;a,b) variates where b is an integer the following logic was used [Ref. 3]. Assume Y is B(y;a,b).

Define $U = -\ln Y$.

Then the moment generating function

$$M_U\ t = E[e^{tU}] = E[e^{-\ln Y}] = E[Y^{-t}] \ .$$

Thus

$$E[Y^{-t}] = \frac{1}{B(a,b)} \int_0^1 y^{a-t-1}(1-y)^{b-1}dy \ ,$$

$$= \frac{B(a-t,b)}{B(a,b)} = \frac{\Gamma(a+b)}{\Gamma(a)\cdot\Gamma(b)} \cdot \frac{\Gamma(b)\Gamma(a-t)}{\Gamma(a+b-t)} \ ,$$

$$= \frac{a(a+1)\dots(a+b-1)}{(a-t)(a+1-t)\dots(a+b-1-t)}$$

$$= \prod_{j=a}^{a+b-1} \left(\frac{1}{1-\frac{t}{j}}\right) \ .$$

Since $\left(\frac{1}{1-\frac{t}{\lambda}}\right)$

is the moment generating function for an exponential random variable with parameter λ, then U is the sum of b independent exponential random variables with failure rates a, a+1, ..., a+b-1,

i.e., $$U = \sum_{j=0}^{b-1} U_j \quad \text{where } U_j \text{ is exp } (a+j).$$

Since $Y = e^{-U}$, random beta variates can be generated with the use of existing exponential generators. For the simulation, where the posterior distributions on R_i were $B3_i(r;a_i,b_i)$ and the b_i's were integers, values for RS were obtained by using the following formula:

$$-\ln RS = \sum_{i=1}^{k} \sum_{j=0}^{b_i-1} U_{ij} ,$$

where

$$U_{ij} \text{ is exponential } (a_i+j).$$

In the case where the b parameter of the beta density was noninteger, random beta variates were generated by using Monte Carlo rejection techniques [Ref. 4]. Since this investigation was concerned with small values of the b parameter the Monte Carlo technique had to be modified slightly. The procedure that was used is outlined below.

1.	Compute P	$P=\text{Prob}(X \leq 0.999)$ where X is $B(x;a,b)$.
2.	Set H	H=Max value of $f_X(x)$ for $0 \leq x \leq 0.999$.
3.	Generate R	R=Random U(0,1) number.
4.	If $R > P$	Then $X_i=1.0$, Go to 3.
5.	Generate R1	R1=Random U(0,1) number.
6.	If $R1 > 0.999$	Then Go to 5.
7.	Generate R2	R2=Random U(0,1) number.
8.	If $(R2 \cdot H \leq f_X(R1)$	Then $X_i=R1$, Go to 3.
9.	Go to 5.	

This procedure was repeated until 500 random beta variates were realized for each posterior density.

C. MODIFICATION OF SUBROUTINE BDTR

Subroutine BDTR [Ref. 5] was used in the simulation to convert beta priors and to compute beta densities in the generation of random beta variates using Monte Carlo techniques. The subroutine BDTR computes the probability that the random variable U is less than or equal to x, where U is distributed according to the beta (u;a,b) distribution and $0 \le x \le 1$. The value of the density at x is also computed. In order for the computations to be valid the sum of the parameters must be greater than or equal to one. In the subroutine BDTR this condition is met by restricting both parameters to values greater than or equal to 0.5. The subroutine was modified so that smaller values of the b parameter could be used since the a parameter was always greater than or equal to one. Two additional variables were added to the parameter list of the subroutine which increased the efficiency of the simulation by acting as flags and saving values when the subroutine was being called during the generation of random beta variates. All modifications to the subroutine are shown in the listing of the computer program.

IV. RESULTS

The results of the simulation are given in tables for each of the three beta priors that were used. The number of failures was the total number of failures in all of the components. For n=50, and in a similar manner where n=75 and n=100, the failures were assigned as follows:

Failures	Successes	
0	s_i = 50,	$i=1,2,\ldots,k$
1	s_i = 49,	$i=1$
	= 50,	$i=2,3,\ldots,k$
2	s_i = 49,	$i=1,2$
	= 50,	$i=3,4,\ldots,k$
3	s_i = 48,	$i=1$
	= 49,	$i=2$
	= 50,	$i=3,4,\ldots,k$
4	s_i = 48,	$i=1,2$
	= 50,	$i=3,4,\ldots,k$
5	s_i = 48,	$i=1,2$
	= 49,	$i=3$
	= 50,	$i=4,5,\ldots,k$
6	s_i = 48,	$i=1,2$
	= 49,	$i=3,4$
	= 50,	$i=5,6,\ldots,k$

The beta priors for the Bayesian cases were as follows:

Case 1	B1(r;a,b)		
Case 2	B21(r;a,1)	$a = \min(0.75n, a^*)$	n=50,75,100
Case 3	B22(r;a,1)	$a = \min(1.0\,n, a^*)$	n=50,75,100
Case 4	B23(r;a,1)	$a = \min(1.5\,n, a^*)$	n=50,75,100

The appropriate values for B1(r;a,b) and B2(r;a*,1) are given at the top of each table.

90% Lower Confidence Limits on RS

B1(r;5.0,0.03) converts to B2(r;596.8,1)

Fifty tests for each component

k	Failures	Classical Case	Bayesian Case 1	Case 2	Case 3	Case 4
40	0	0.9540	0.9532	0.5751	0.6162	0.6789
40	1	0.9222	0.9272	0.5739	0.6151	0.6779
40	2	0.8936	0.9038	0.5599	0.6020	0.6663
40	3	0.8664	0.8827	0.5541	0.5966	0.6116
40	4	0.8401	0.8647	0.5493	0.5920	0.6575
40	5	0.8145	0.8410	0.5429	0.5861	0.6523
40	6	0.7894	0.8247	0.5338	0.5775	0.6446
30	0	0.9540	0.9616	0.6547	0.6903	0.7434
30	1	0.9222	0.9345	0.6467	0.6829	0.7371
30	2	0.8936	0.9119	0.6384	0.6752	0.7305
30	3	0.8664	0.8897	0.6258	0.6636	0.7205
30	4	0.8401	0.8696	0.6247	0.6627	0.7196
30	5	0.8145	0.8491	0.6179	0.6563	0.7140
30	6	0.7894	0.8307	0.6075	0.6467	0.7057
20	0	0.9540	0.9731	0.7452	0.7731	0.8139
20	1	0.9222	0.9398	0.7366	0.7653	0.8074
20	2	0.8936	0.9159	0.7277	0.7572	0.8006
20	3	0.8664	0.8950	0.7192	0.7495	0.7940
20	4	0.8401	0.8754	0.7074	0.7388	0.7850
20	5	0.8145	0.8549	0.7012	0.7331	0.7802
20	6	0.7894	0.8348	0.6924	0.7251	0.7734

90% Lower Confidence Limits on RS

B1(r;5.0,0.03) converts to B2(r;596.8,1)

Fifty tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
10	0	0.9540	0.9840	0.8478	0.8654	0.8908
10	1	0.9222	0.9474	0.8413	0.8597	0.8861
10	2	0.8936	0.9217	0.8306	0.8501	0.8782
10	3	0.8664	0.9023	0.8156	0.8368	0.8673
10	4	0.8401	0.8817	0.8069	0.8289	0.8607
10	5	0.8145	0.8576	0.7949	0.8182	0.8538
10	6	0.7894	0.8399	0.7822	0.8067	0.8423
5	0	0.9540	0.9924	0.9163	0.9264	0.9407
5	1	0.9222	0.9521	0.9023	0.9140	0.9306
5	2	0.8936	0.9259	0.8850	0.8986	0.9180
5	3	0.8664	0.9062	0.8763	0.8910	0.9119
5	4	0.8401	0.8833	0.8637	0.8798	0.9027
5	5	0.8145	0.8616	0.8466	0.8645	0.8902
5	6	0.7894	0.8435	0.8400	0.8587	0.8856

90% Lower Confidence Limits on RS

B1(r;5.0,0.03) converts to B2(r;596.8,1)

Seventy-five tests for each component

		Classical		Bayesian		
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
40	0	0.9693	0.9701	0.6915	0.7242	0.7724
40	1	0.9481	0.9496	0.6906	0.7233	0.7717
40	2	0.9290	0.9366	0.6794	0.7130	0.7629
40	3	0.9109	0.9135	0.6747	0.7088	0.7593
40	4	0.8934	0.9032	0.6708	0.7051	0.7561
40	5	0.8763	0.8865	0.6658	0.7006	0.7523
40	6	0.8596	0.8721	0.6582	0.6936	0.7463
30	0	0.9693	0.9756	0.7540	0.7811	0.8206
30	1	0.9481	0.9552	0.7478	0.7755	0.8160
30	2	0.9290	0.9402	0.7415	0.7697	0.8111
30	3	0.9109	0.9186	0.7318	0.7610	0.8038
30	4	0.8934	0.9062	0.7310	0.7602	0.8031
30	5	0.8763	0.8920	0.7256	0.7553	0.7989
30	6	0.8596	0.8780	0.7175	0.7480	0.7927
20	0	0.9693	0.9803	0.8219	0.8423	0.8717
20	1	0.9481	0.9591	0.8157	0.8367	0.8671
20	2	0.9290	0.9435	0.8091	0.8309	0.8622
20	3	0.9109	0.9235	0.8028	0.8251	0.8575
20	4	0.8934	0.9093	0.7941	0.8174	0.8510
20	5	0.8763	0.8958	0.7895	0.8132	0.8476
20	6	0.8596	0.8804	0.7829	0.8073	0.8427

90% Lower Confidence Limits on RS

B1(r;5.0,0.03) converts to B2(r;596.8,1)

Seventy-five tests for each component

		Classical		Bayesian		
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
10	0	0.9693	0.9891	0.8957	0.9082	0.9258
10	1	0.9481	0.9647	0.8912	0.9041	0.9225
10	2	0.9290	0.9474	0.8836	0.8974	0.9171
10	3	0.9109	0.9267	0.8732	0.8882	0.9095
10	4	0.8934	0.9138	0.8669	0.8825	0.9049
10	5	0.8763	0.9001	0.8584	0.8750	0.8987
10	6	0.8596	0.8836	0.8492	0.8668	0.8920
5	0	0.9693	0.9940	0.9434	0.9503	0.9600
5	1	0.9481	0.9677	0.9338	0.9418	0.9532
5	2	0.9290	0.9504	0.9218	0.9313	0.9447
5	3	0.9109	0.9304	0.9160	0.9261	0.9405
5	4	0.8934	0.9159	0.9072	0.9183	0.9342
5	5	0.8763	0.9002	0.8951	0.9077	0.9255
5	6	0.8596	0.8850	0.8908	0.9038	0.9224

90% Lower Confidence Limits on RS

B1(r;5.0,0.03) converts to B2(r;596.8,1)

One hundred tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
40	0	0.9770	0.9749	0.7583	0.7850	0.8239
40	1	0.9611	0.9586	0.7576	0.7843	0.8234
40	2	0.9458	0.9481	0.7483	0.7759	0.8163
40	3	0.9332	0.9380	0.7445	0.7725	0.8134
40	4	0.9200	0.9266	0.7412	0.7695	0.8109
40	5	0.9073	0.9106	0.7372	0.7658	0.8079
40	6	0.8947	0.9019	0.7308	0.7601	0.8030
30	0	0.9770	0.9817	0.8091	0.8308	0.8622
30	1	0.9611	0.9641	0.8042	0.8264	0.8585
30	2	0.9458	0.9514	0.7991	0.8218	0.8547
30	3	0.9332	0.9417	0.7913	0.8149	0.8490
30	4	0.9200	0.9270	0.7906	0.8142	0.8484
30	5	0.9073	0.9133	0.7862	0.8102	0.8451
30	6	0.8947	0.9070	0.7797	0.8043	0.8402
20	0	0.9770	0.9861	0.8632	0.8793	0.9022
20	1	0.9611	0.9685	0.8583	0.8749	0.8986
20	2	0.9458	0.9541	0.8532	0.8703	0.8948
20	3	0.9332	0.9446	0.8481	0.9658	0.8911
20	4	0.9200	0.9304	0.8413	0.8596	0.8861
20	5	0.9073	0.9183	0.8337	0.8564	0.8834
20	6	0.8947	0.9090	0.8324	0.8518	0.8796

90% Lower Confidence Limits on RS

B1(r;5.0,0.03) converts to B2(r;596.8,1)

One hundred tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
10	0	0.9770	0.9918	0.9207	0.9303	0.9438
10	1	0.9611	0.9727	0.9172	0.9272	0.9413
10	2	0.9458	0.9595	0.9114	0.9220	0.9371
10	3	0.9332	0.9474	0.9034	0.9150	0.9314
10	4	0.9200	0.9336	0.8984	0.9106	0.9278
10	5	0.9073	0.9210	0.8919	0.9047	0.9127
10	6	0.8947	0.9101	0.8847	0.8984	0.9179
5	0	0.9770	0.9956	0.9572	0.9625	0.9699
5	1	0.9611	0.9753	0.9499	0.9560	0.9647
5	2	0.9458	0.9609	0.9408	0.9480	0.9582
5	3	0.9332	0.9504	0.9364	0.9441	0.9551
5	4	0.9200	0.9344	0.9296	0.9382	0.9503
5	5	0.9073	0.9222	0.9204	0.9300	0.9436
5	6	0.8947	0.9121	0.9171	0.9271	0.9413

90% Lower Confidence Limits on RS

B1(r;5.0,0.05) converts to B2(r;136.8,1)

Fifty tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
40	0	0.9540	0.9344	0.5751	0.6162	0.6789
40	1	0.9222	0.9085	0.5739	0.6151	0.6779
40	2	0.8936	0.8844	0.5599	0.6020	0.6663
40	3	0.8664	0.8715	0.5541	0.5966	0.6116
40	4	0.8401	0.8459	0.5493	0.5920	0.6575
40	5	0.8145	0.8265	0.5429	0.5861	0.6523
40	6	0.7894	0.8091	0.5338	0.5775	0.6446
30	0	0.9540	0.9441	0.6547	0.6903	0.7434
30	1	0.9222	0.9203	0.6467	0.6829	0.7371
30	2	0.8936	0.8918	0.6384	0.6752	0.7305
30	3	0.8664	0.8758	0.6258	0.6636	0.7205
30	4	0.8401	0.8542	0.6247	0.6627	0.7196
30	5	0.8145	0.8318	0.6179	0.6563	0.7140
30	6	0.7894	0.8153	0.6075	0.6467	0.7057
20	0	0.9540	0.9608	0.7452	0.7731	0.8139
20	1	0.9222	0.9329	0.7366	0.7653	0.8074
20	2	0.8936	0.9072	0.7277	0.7572	0.8006
20	3	0.8664	0.8874	0.7192	0.7495	0.7940
20	4	0.8401	0.8607	0.7074	0.7388	0.7850
20	5	0.8145	0.8385	0.7012	0.7331	0.7802
20	6	0.7894	0.8243	0.6924	0.7251	0.7734

90% Lower Confidence Limits on RS

B1(r;5.0,0.05) converts to B2(r;136.8,1)

Fifty tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
10	0	0.9540	0.9749	0.8478	0.8654	0.8908
10	1	0.9222	0.9452	0.8413	0.8597	0.8861
10	2	0.8936	0.9142	0.8306	0.8501	0.8782
10	3	0.8664	0.8934	0.8156	0.8368	0.8673
10	4	0.8401	0.8672	0.8069	0.8289	0.8607
10	5	0.8145	0.8469	0.7949	0.8182	0.8538
10	6	0.7894	0.8313	0.7822	0.8067	0.8423
5	0	0.9540	0.9866	0.9163	0.9264	0.9407
5	1	0.9222	0.9521	0.9023	0.9140	0.9306
5	2	0.8936	0.9234	0.8850	0.8986	0.9180
5	3	0.8664	0.9027	0.8763	0.8910	0.9119
5	4	0.8401	0.8725	0.8637	0.8798	0.9027
5	6	0.7894	0.8384	0.8400	0.8587	0.8856

90% Lower Confidence Limits on RS

B1(r;5.0,0.05) converts to B2(r;136.8,1)

Seventy-five tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
40	0	0.9693	0.9533	0.6915	0.7242	0.7724
40	1	0.9481	0.9388	0.6906	0.7233	0.7717
40	2	0.9290	0.9209	0.6794	0.7130	0.7629
40	3	0.9109	0.9046	0.6747	0.7088	0.7593
40	4	0.8934	0.8943	0.6708	0.7051	0.7561
40	5	0.8763	0.8769	0.6658	0.7006	0.7523
40	6	0.8596	0.8601	0.6582	0.6936	0.7463
30	0	0.9693	0.9615	0.7540	0.7811	0.8206
30	1	0.9481	0.9449	0.7478	0.7755	0.8160
30	2	0.9290	0.9310	0.7415	0.7697	0.8111
30	3	0.9109	0.9099	0.7318	0.7610	0.8038
30	4	0.8934	0.8988	0.7310	0.7602	0.8031
30	5	0.8763	0.8808	0.7256	0.7553	0.7989
30	6	0.8596	0.8684	0.7175	0.7480	0.7927
20	0	0.9693	0.9710	0.8219	0.8423	0.8717
20	1	0.9481	0.9526	0.8157	0.8367	0.8671
20	2	0.9290	0.9367	0.8091	0.8309	0.8622
20	3	0.9109	0.9172	0.8028	0.8251	0.8575
20	4	0.8934	0.9057	0.7941	0.8174	0.8510
20	5	0.8763	0.8885	0.7895	0.8132	0.8476
20	6	0.8596	0.8770	0.7829	0.8073	0.8427

90% Lower Confidence Limits on RS

B1(r;5.0,0.05) converts to B2(r;136.8,1)

Seventy-five tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
10	0	0.9693	0.9826	0.8957	0.9082	0.9258
10	1	0.9481	0.9597	0.8912	0.9041	0.9225
10	2	0.9290	0.9421	0.8836	0.8974	0.9171
10	3	0.9109	0.9245	0.8732	0.8882	0.9095
10	4	0.8934	0.9101	0.8669	0.8825	0.9049
10	5	0.8763	0.8959	0.8584	0.8750	0.8987
10	6	0.8596	0.8819	0.8492	0.8668	0.8920
5	0	0.9693	0.9892	0.9431	0.9503	0.9600
5	1	0.9481	0.9657	0.9338	0.9418	0.9532
5	2	0.9290	0.9476	0.9218	0.9313	0.9447
5	3	0.9109	0.9286	0.9160	0.9261	0.9405
5	4	0.8934	0.9160	0.9072	0.9183	0.9342
5	5	0.8763	0.8972	0.8951	0.9077	0.9255
5	6	0.8596	0.8842	0.8908	0.9038	0.9224

90% Lower Confidence Limits on RS

B1(r;5.0,0.05) converts to B2(r;136.8,1)

One hundred tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
40	0	0.9770	0.9649	0.7583	0.7850	0.8151
40	1	0.9611	0.9507	0.7576	0.7843	0.8145
40	2	0.9458	0.9373	0.7483	0.7759	0.8071
40	3	0.9332	0.9277	0.7445	0.7695	0.8015
40	4	0.9200	0.9148	0.7412	0.7695	0.8015
40	5	0.9073	0.9086	0.7372	0.7658	0.7983
40	6	0.8947	0.8945	0.7308	0.7691	0.7932
30	0	0.9770	0.9721	0.8091	0.8391	0.8308
30	1	0.9611	0.9579	0.8042	0.8264	0.8513
30	2	0.9458	0.9444	0.7991	0.8218	0.8473
30	3	0.9332	0.9336	0.7913	0.8149	0.8413
30	4	0.9200	0.9207	0.7906	0.8142	0.8406
30	5	0.9073	0.9120	0.7862	0.8102	0.8372
30	6	0.8947	0.9011	0.7797	0.8043	0.8313
20	0	0.9770	0.9794	0.8632	0.8793	0.8970
20	1	0.9611	0.9641	0.8583	0.8749	0.8932
20	2	0.9458	0.9496	0.8532	0.8703	0.8893
20	3	0.9332	0.9395	0.8481	0.8658	0.8854
20	4	0.9200	0.9270	0.8413	0.8596	0.8801
20	5	0.9073	0.9172	0.8377	0.8564	0.8774
20	6	0.8947	0.9063	0.8324	0.8518	0.8733

90% Lower Confidence Limits on RS

B1(r;5.0,0.05) converts to B2(r;136.8,1)

One hundred tests for each component

		Classical		Bayesian		
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
10	0	0.9770	0.9871	0.9207	0.9303	0.9408
10	1	0.9611	0.9696	0.9172	0.9272	0.9382
10	2	0.9458	0.9566	0.9114	0.9220	0.9338
10	3	0.9332	0.9457	0.9034	0.9150	0.9277
10	4	0.9200	0.9327	0.8984	0.9106	0.9240
10	5	0.9073	0.9219	0.8919	0.9047	0.9190
10	6	0.8947	0.9091	0.8847	0.8984	0.9135
5	0	0.9770	0.9935	0.9572	0.9625	0.9682
5	1	0.9611	0.9739	0.9499	0.9560	0.9627
5	2	0.9458	0.9599	0.9408	0.9480	0.9560
5	3	0.9332	0.9478	0.9364	0.9441	0.9526
5	4	0.9200	0.9359	0.9296	0.9382	0.9476
5	5	0.9073	0.9271	0.9204	0.9300	0.9406
5	6	0.8947	0.9114	0.9171	0.9271	0.9381

90% Lower Confidence Limits on RS

B1(r;5.0,0.10) converts to B2(r;39.5,1)

Fifty tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
40	0	0.9540	0.8825	0.5751	0.5821	Same as Case 3
40	1	0.9222	0.8639	0.5739	0.5809	
40	2	0.8936	0.8407	0.5599	0.5671	
40	3	0.8664	0.8230	0.5541	0.5613	
40	4	0.8401	0.8028	0.5493	0.5566	
40	5	0.8145	0.7872	0.5429	0.5503	
40	6	0.7894	0.7666	0.5338	0.5412	
30	0	0.9540	0.9055	0.6547	0.6608	
30	1	0.9222	0.8842	0.6467	0.6529	
30	2	0.8936	0.8619	0.6384	0.6447	
30	3	0.8664	0.8423	0.6258	0.6322	
30	4	0.8401	0.8181	0.6247	0.6313	
30	5	0.8145	0.8013	0.6179	0.6179	
30	6	0.7894	0.7833	0.6075	0.6142	
20	0	0.9540	0.9315	0.7452	0.7500	
20	1	0.9222	0.9083	0.7366	0.7416	
20	2	0.8936	0.8864	0.7277	0.7328	
20	3	0.8664	0.8633	0.7192	0.7244	
20	4	0.8401	0.8414	0.7074	0.7128	
20	5	0.8145	0.8238	0.7012	0.7067	
20	6	0.7894	0.8040	0.6924	0.6981	

90% Lower Confidence Limits on RS

B1(r;5.0,0.10) converts to B2(r;39.5,1)

Fifty tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
10	0	0.9540	0.9600	0.8478	0.8508	Same as Case 3
10	1	0.9222	0.9360	0.8413	0.8445	
10	2	0.8936	0.9084	0.8306	0.8340	
10	3	0.8664	0.8832	0.8156	0.8193	
10	4	0.8401	0.8645	0.8069	0.8107	
10	5	0.8145	0.8456	0.7949	0.7990	
10	6	0.7894	0.8230	0.7822	0.7865	
5	0	0.9540	0.9730	0.9163	0.9181	
5	1	0.9222	0.9446	0.9023	0.9044	
5	2	0.8936	0.9207	0.8850	0.8873	
5	3	0.8664	0.9127	0.8763	0.8741	
5	4	0.8401	0.8715	0.8637	0.8665	
5	5	0.8145	0.8531	0.8466	0.8497	
5	6	0.7894	0.8304	0.8400	0.8433	

90% Lower Confidence Limits on RS

B1(r;5.0,0.10) converts to B2(r;39.5,1)

Seventy-five tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
40	0	0.9693	0.9193	0.6551	Same as Case 2	Same as Case 2
40	1	0.9481	0.9041	0.6541		
40	2	0.9290	0.8918	0.6419		
40	3	0.9109	0.8735	0.6369		
40	4	0.8934	0.8641	0.6326		
40	5	0.8763	0.8520	0.6271		
40	6	0.8596	0.8365	0.6190		
30	0	0.9693	0.9372	0.7234		
30	1	0.9481	0.9201	0.7166		
30	2	0.9290	0.9044	0.7096		
30	3	0.9101	0.8884	0.6990		
30	4	0.8934	0.8782	0.6981		
30	5	0.8763	0.8626	0.6922		
30	6	0.8596	0.8509	0.6834		
20	0	0.9693	0.9541	0.7986		
20	1	0.9481	0.9382	0.7916		
20	2	0.9290	0.9221	0.7843		
20	3	0.9109	0.9033	0.7773		
20	4	0.8934	0.8925	0.7677		
20	5	0.8763	0.8800	0.7625		
20	6	0.8596	0.8616	0.7552		

90% Lower Confidence Limits on RS

B1(r;5.0,0.10) converts to B2(r;39.5,1)

Seventy-five tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
10	0	0.9693	0.9718	0.8814	Same as Case 2	Same as Case 2
10	1	0.9481	0.9520	0.8763		
10	2	0.9290	0.9369	0.8677		
10	3	0.9109	0.9167	0.8559		
10	4	0.8934	0.9057	0.8488		
10	5	0.8763	0.8912	0.8393		
10	6	0.8596	0.8758	0.8290		
5	0	0.9693	0.9837	0.9354		
5	1	0.9481	0.9604	0.9244		
5	2	0.9290	0.9446	0.9108		
5	3	0.9190	0.9257	0.9042		
5	4	0.8934	0.9122	0.8942		
5	5	0.8763	0.8989	0.8806		
5	6	0.8596	0.8825	0.8756		

90% Lower Confidence Limits on RS

B1(r;5.0,0.10) converts to B2(r;39.5,1)

One hundred tests for each component

		Classical	Bayesian			
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
40	0	0.9770	0.9411	0.7067	Same as Case 2	Same as Case 2
40	1	0.9611	0.9266	0.7058		
40	2	0.9458	0.9130	0.6950		
40	3	0.9332	0.9045	0.6905		
40	4	0.9200	0.8925	0.6867		
40	5	0.9073	0.8855	0.6819		
40	6	0.8947	0.8743	0.6747		
30	0	0.9770	0.9495	0.7666		
30	1	0.9611	0.9372	0.7608		
30	2	0.9458	0.9258	0.7547		
30	3	0.9332	0.9164	0.7454		
30	4	0.9200	0.9034	0.7446		
30	5	0.9073	0.8960	0.7394		
30	6	0.8947	0.8838	0.7317		
20	0	0.9770	0.9652	0.8315		
20	1	0.9611	0.9517	0.8255		
20	2	0.9458	0.9379	0.8193		
20	3	0.9332	0.9276	0.8132		
20	4	0.9200	0.9175	0.8050		
20	5	0.9073	0.9074	0.8006		
20	6	0.8947	0.8969	0.7943		

90% Lower Confidence Limits on RS

B1(r;5.0,0.10) converts to B2(r;39.5,1)

One hundred tests for each component

		Classical		Bayesian		
k	Failures	Case	Case 1	Case 2	Case 3	Case 4
10	0	0.9770	0.9797	0.9016	Same as Case 2	Same as Case 2
10	1	0.9611	0.9619	0.8973		
10	2	0.9458	0.9505	0.8901		
10	3	0.9332	0.9409	0.8802		
10	4	0.9200	0.9280	0.8742		
10	5	0.9073	0.9148	0.8661		
10	6	0.8947	0.9062	0.8575		
5	0	0.9770	0.9867	0.9466		
5	1	0.9611	0.9696	0.9376		
5	2	0.9458	0.9560	0.9262		
5	3	0.9332	0.9450	0.9207		
5	4	0.9200	0.9319	0.9124		
5	5	0.9073	0.9217	0.9010		
5	6	0.8947	0.9112	0.8969		

V. CONCLUSIONS

The results of the simulation indicate that the classical approach to system reliability usually yields higher values of lower confidence limits than the Bayesian approach. When there are 10 or more components, only the beta priors with b parameters less than one provided higher values of lower confidence limits than the classical approach. When using the Bayesian approach, the effect of each prior density assumption is to obtain a more optimistic posterior density on each component reliability. However, there is some probability mass that is still assigned to small values of each R_i. Thus, as the number of components increase the probability of small values of RS increases since RS is a product of the R_i's. Therefore, as the number of components increase, the effect of the prior density assumptions tends to diminish.

APPENDIX A

BETA DENSITY CONVERSIONS

The following table lists values of a*, where a* is such that the 10th percentile point of a B(x;a,b) distribution is the same as the 10th percentile point of a B(x;a*,1) distribution.

Table of a*'s

b	a=1	a=5	a=10
1.00	0.0	5.0	10.0
0.90	1.05	5.36	10.77
0.80	1.10	5.81	11.72
0.70	1.17	6.37	12.91
0.60	1.26	7.09	14.46
0.50	1.39	8.11	16.61
0.45	1.47	8.77	18.02
0.40	1.57	9.61	19.78
0.35	1.71	10.68	22.06
0.30	1.89	12.14	25.13
0.25	2.16	14.23	29.55
0.20	2.58	17.56	36.57
0.18	2.82	19.52	40.70
0.16	3.16	22.11	46.16
0.15	3.37	23.75	49.59
0.14	3.61	25.69	53.69
0.12	4.28	30.96	64.78

Table of a*'s (Continued)

b	a=1	a=5	a=10
0.10	5.37	39.46	82.67
0.09	6.20	45.98	96.38
0.08	7.38	55.24	115.87
0.07	9.17	69.29	145.44
0.06	12.14	92.61	194.51
0.05	17.76	136.82	287.54
0.04	30.91	240.42	505.53
0.03	76.02	596.85	1256.03
0.02	445.64	3528.06	7438.38

COMPUTER PROGRAM

```
C       ..................................................................
C
C       MAIN PROGRAM
C
C       THE PURPOSE OF THIS PROGRAM IS TO SIMULATE A SERIES
C       SYSTEM OF  K COMPONENTS AND TO COMPUTE BAYESIAN LOWER
C       CONFIDENCE LIMITS OF SYSTEM RELIABILITY.
C
C       INPUTS -- NUMBER OF COMPONENTS
C              -- BETA PRIORS
C              -- TEST DATA
C       ..................................................................
C
        DOUBLE PRECISION DLBETA,XX,PP,AA
        DIMENSION  RS1(500),RS21(500),RS22(500),RS23(500),
       1NI(50),NSI(50),BETA1(50,2),BETA2(50,2),BETA21(50,2),
       2BETA22(50,2),BETA23(50,2),BETA11(50,2),XB(40,500),
       3BETA31(50,2),BETA32(50,2),BETA33(50,2)
        DATA RS1/500 * 1.0/
C
C       INITIALIZE PROGRAM
C
        H = 26.4811
        DD1 = 47.7160
        DD = 19.8219
        IRUN = 0
        KTIME = 0
        PWANT = 0.10
        IX = 36941
        IFIRST = IX
C
C       READ NUMBER OF COMPONENTS IN SYSTEM
C
        READ (5,9000) K
   9000 FORMAT(I3)
C
C       READ INITIAL BETA PRIORS
C
        READ (5,9001) ((BETA1(I,J),J=1,2),I=1,K)
   9001 FORMAT (2F10.4)
C
C       READ TEST DATA--NUMBER OF TRIALS AND SUCCESSES
C
        READ(5,9002)((NI(I),NSI(I),I=1,K)
   9002 FORMAT(2I10)
C
C       CONVERT INITIAL BETA PRIORS TO BETA(A,1) DENSITIES
C
   1000 CONTINUE
        DO 1099 I = 1,K
        B = BETA1(I,2)
        A = BETA1(I,1)
        IF(I.EQ.1) GO TO 1011
        IF((A.EQ.BETA1(I-1,1)).AND.(B.EQ.BETA1(I-1,2))) GO TO
       11050
   1011 CONTINUE
        X = 0.0
   1015 X1 = 0.1
        IFLAG = 0
        X = X + X1
        IF(X.GT.1.0) GO TO 1030
   1019 IP = 1
        IER = 1
   1020 CALL BDTR(X,A,B,P,D,IER,IP,DLBETA)
        IF (P.GT.PWANT ) GO TO 1030
        X = X + X1
        IF(X.GT.1.0) GO TO 1030
        GO TO 1019
```

```
 1030 IFLAG = IFLAG + 1
      IF (IFLAG.EQ.6) GO TO 1040
      X = X - X1
      X1 = X1 * 0.1
      GO TO 1019
 1040 CONTINUE
      PP = DBLE(P)
      XX = DBLE(X)
      AA = DLOG(PP)/DLOG(XX)
      A4B1 = SNGL (AA)
      BETA2(I,1) = A4B1
      BETA2(I,2) = 1.0
      GO TO 1099
 1050 BETA2(I,1) = BETA2(I-1,1)
      BETA2(I,2) = 1.0
 1099 CONTINUE
C
C     ADJUST CONVERTED BETA PRIORS
C
      DO 1100 I=1,K
      BETA21(I,2) = 1.0
      BETA22(I,2) = 1.0
      BETA23(I,2) = 1.0
      BETA21(I,1) = 0.75 * NI(I)
      BETA22(I,1) = 1.0  * NI(I)
      BETA23(I,1) = 1.5  * NI(I)
      IF(BETA21(I,1).GT.BETA2(I,1)) BETA21(I,1) = BETA2(I,1)
      IF(BETA22(I,1).GT.BETA2(I,1)) BETA22(I,1) = BETA2(I,1)
      IF(BETA23(I,1).GT.BETA2(I,1)) BETA23(I,1) = BETA2(I,1)
 1100 CONTINUE
C
C     COMPUTE BETA PORTERIORS FROM ADJUSTED PRIORS
C
 1101 CONTINUE
      DO 1110 I = 1,K
      BETA31(I,1) = BETA21(I,1) + NSI(I)
      BETA32(I,1) = BETA22(I,1) + NSI(I)
      BETA33(I,1) = BETA23(I,1) + NSI(I)
      BETA31(I,2) = BETA21(I,2) + NI(I) - NSI(I)
      BETA32(I,2) = BETA22(I,2) + NI(I) - NSI(I)
      BETA33(I,2) = BETA23(I,2) + NI(I) - NSI(I)
 1110 CONTINUE
C
C     GENERATE RANDOM BETA VARIATES AND REALIZE VALUES OF RS
C
 1111 CONTINUE
      IX = IFIRST
      DO 1125 K1 = 1,500
      RSLN1 = 0.0
      RSLN2 = 0.0
      RSLN3 = 0.0
      DO 1120 I = 1,K
      IB1 = BETA31(I,2)
      B1 = BETA31(I,1)
      B2 = BETA32(I,1)
      B3 = BETA33(I,1)
      DO 1115 J = 1,IB1
 1114 CALL RANDU(IX,IY,RAN)
      IX = IY
      IF(RAN.EQ.0.0) GO TO 1114
      X = -ALOG(RAN)
      X1 = X/(J-1+B1)
      X2 = X/(J-1+B2)
      X3 = X/(J-1+B3)
      RSLN1 = RSLN1 + X1
      RSLN2 = RSLN2 + X2
      RSLN3 = RSLN3 + X3
 1115 CONTINUE
 1120 CONTINUE
      RS21(K1) = EXP(-RSLN1)
      RS22(K1) = EXP(-RSLN2)
      RS23(K1) = EXP(-RSLN3)
```

```
1125 CONTINUE
     NPASS = 499
     DO 1150 I = 1,NPASS
     NSTOP = NPASS - I + 1
     DO 1150 J = 1,NSTOP
     IF(RS21(J).LE.RS21(J+1)) GO TO 1130
     TEMP = RS21(J)
     RS21(J) = RS21(J+1)
     RS21(J+1) = TEMP
1130 CONTINUE
     IF(RS22(J).LE.RS22(J+1)) GO TO 1135
     TEMP = RS22(J)
     RS22(J) = RS22(J+1)
     RS22(J+1) = TEMP
1135 CONTINUE
     IF(RS23(J).LE.RS23(J+1)) GO TO 1150
     TEMP = RS23(J)
     RS23(J) = RS23(J+1)
     RS23(J+1) = TEMP
1150 CONTINUE
2000 CONTINUE
     IF (K.LT.40) GO TO 2201
     IR = IFIRST
C
C    COMPUTE BETA  POSTERIORS USING THE INITAL BETA PRIORS
C
     DO 2010 I = 1, K
     BETA11(I,1) = BETA1(I,1) + NSI(I)
     BETA11(I,2) = BETA1(I,2) + NI(I) - NSI(I)
2010 CONTINUE
C
C    GENERATE RANDOM BETA VARIATES AND REALIZE VALUES OF RS
C
     IF(IRUN.EQ.0) KRUN = K
     IF(IRUN.EQ.1) KRUN = 1
     IF(IRUN.GT.1) KRUN = 2
     IF(IRUN.EQ.5) KRUN = 3
     IF(IRUN.EQ.6) KRUN = 4
     DO 2200 I = 1,KRUN
     X = 0.999
     A = BETA11(I,1)
     B = BETA11(I,2)
     KK = 0
     IF(B.GE.2.0) GO TO 2020
     IF(B.GE.1.0) KK = 1
     GO TO 2025
2020 KK = 2
2025 CONTINUE
     IP = 1
     IER = 1
     CALL BDTR(X,A,B,P,D,IER,IP,DLBETA)
     IP = 0
     U = P
     IF(KK.EQ.0) HT = H
     IF(KK.EQ.1) HT = DD1
     IF(KK.EQ.2) HT = DD
     X = 0.9
     IER = 1
     CALL BDTR(X,A,B,P,D,IER,IP,DLBETA)
     H2 = D
     X = 0.99
     IER = 1
     CALL BDTR(X,A,B,P,D,IER,IP,DLBETA)
     H3 = D
     DO 2120 K1 = 1,500
     CALL RANDU(IX,IY,RAN1)
     IX = IY
     IF (RAN1 - U) 1,1,60
   1 CALL RANDU(IX,IY,RAN1)
     IX = IY
     IF (RAN1 - 0.999) 6,6,1
   6 IF(RAN1 - 0.9) 20,20,30
```

```
   10 CALL RANDU(IX,IY,RAN1)
      IX = IY
      IF (RAN1 - 0.999) 11,11,10
   11 IF (RAN1 - 0.9) 20,20,30
   20 CALL RANDU(IX,IY,RAN2)
      IX = IY
      Y = RAN1 * (H2/0.9)
      IF ((RAN2 * HT)- Y) 40,10,10
   30 CALL RANDU(IX,IY,RAN2)
      IX = IY
      IF(KK.EQ.2) GO TO 40
      IF(RAN1 - 0.99) 31,31,40
   31 CONTINUE
      Y = (RAN1 - 0.9) * ((H3 - H2)/0.09) + H2
      IF ((RAN2 * HT)- Y) 40,10,10
   40 X = RAN1
      CALL BDTR(X,A,B,P,D,IER,IP,DLBETA)
      H1 = D
      IF((RAN2 * HT) - H1) 50,1,1
   50 XI = RAN1
      XB(I,K1) = XI
      GO TO 2120
   60 XI = 1.0
      XB(I,K1) = XI
 2120 CONTINUE
 2200 CONTINUE
 2201 CONTINUE
      DO 2250 I = 1,500
      RS1(I) = 1.0
      DO 2240 J = 1,K
      RS1(I) = RS1(I) * XB(J,I)
 2240 CONTINUE
 2250 CONTINUE
      NPASS = 499
      DO 2300 L = 1,NPASS
      NSTOP = NPASS - L + 1
      DO 2300 J = 1,NSTOP
      IF(RS1(J).LE.RS1(J+1)) GO TO 2300
      TEMP = RS1(J)
      RS1(J) = RS1(J+1)
      RS1(J+1) = TEMP
 2300 CONTINUE
C
C     WRITE RESULTS OF SIMULATION
C
      WRITE(6,9300)
 9300 FORMAT('1','COMPONENT  BETA PRIOR  CONVERTED TO B(A,',
     1'1)  TESTS  SUCCESSES',/)
      DO 2900 I = 1,K
      WRITE(6,9301) I,BETA1(I,1),BETA1(I,2),BETA2(I,1),
     1BETA2(I,2),NI(I),NSI(I)
 9301 FORMAT(5X,I2,'     B(',F4.1,',',F4.2,')    B(',F6.1,',',
     1F3.1,')            ',I3,8X,I3,/)
 2900 CONTINUE
 3000 CONTINUE
 9200 FORMAT ('1')
 3100 CONTINUE
 9010 FORMAT('0','CASE ',I1,/,
     1' 10TH PERCENTILE POINT OF RS = ',F7.4,//,
     2' 20TH PERCENTILE POINT OF RS = ',F7.4,//)
      II = 2
      WRITE(6,9010) II,RS21(50),RS21(100)
      II = 3
      WRITE(6,9010) II,RS22(50),RS22(100)
      II = 4
      WRITE(6,9010) II,RS23(50),RS23(100)
C
C     RESET PARAMETERS
C
      KTIME = KTIME + 1
      IF(KTIME.EQ.1) K=30
      IF(KTIME.EQ.2) K=20
```

```
      IF(KTIME.EQ.3) K=10
      IF(KTIME.EQ.4) K=5
      IF(KTIME.EQ.5) GO TO 9950
      GO TO 1111
 9950 CONTINUE
      IRUN = IRUN + 1
      IF(IRUN.EQ.7) GO TO 9999
      READ(5,9002)(NI(I),NSI(I),I=1,5)
      K = 40
      KTIME = 0
      GO TO 1101
 9999 CONTINUE
 9998 FORMAT ('1','RUN COMPLETE',/)
      WRITE (6,9998)
      STOP
      END
C                         SUBROUTINE BDTR
C     ..................................................................
C
C     PURPOSE
C        COMPUTES F(X) = PROBABILITY THAT THE RANDOM VARIBLE
C        U, DISTRIBUTED ACCORDING TO THE BETA DISTRIBUTION
C        WITH PARAMETERS A AND B, IS LESS THAN OR EQUAL TO
C        X.  F(A,B,X), THE ORDINATE OF THE BETA DENSITY AT
C        X IS ALSO COMPUTED.
C
C     USAGE
C        CALL BDTR(X,A,B,P,D,IER)
C
C     DISCRIPTION OF PARAMETERS
C        X    - INPUT SCALAR FOR WHICH P(X) IS COMPUTED.
C        A    - BETA DISTRIBUTION PARAMETER (CONTINUOUS).
C        B    - BETA DISTRIBUTION PARAMETER (CONTINUOUS).
C        P    - OUTPUT PROBABILITY.
C        D    - OUTPUT DENSITY.
C        IER  - RESULTANT ERROR CODE WHERE
C               IER= 0 --- NO ERROR
C               IER=-1,+1  CDTR HAS BEEN CALLED AND AN ERROR
C                          HAS OCCURRED.  SEE CDTR.
C               IER=-2 --- AN INPUT PARAMETER IS INVALID.
C               IER=+2 --- INVALID OUTPUT.
C
C     REMARKS
C        SEE MATEMATICAL DESCRIPTION.
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        DLGAM
C        NDTR
C        CDTR
C
C     METHOD
C        REFER TO R.E. BARGMANN AND S.P. GHOSH, STATISTICAL
C        DISTRIBUTION PROGRAMS FOR A COMPUTER LANGUAGE,
C        IBM RESEARCH REPOST RC-1094, 1963.
C     ..................................................................
C
C     MODIFICATION TO SUBROUTINE BDTR
C
C        USAGE --- CALL BDTR(X,A,B,P,D,IER,IP,DLBETA)
C
C        DISCRIPTION OF ADDITIONAL PARAMETERS
C              IP ------ ACTS AS A FLAG TO ELIMINATE THE
C                        COMPUTATION OF DLBETA.
C              DLBETA -- ADDED TO PARAMETER LIST TO SAVE.
C              IER ----- UTILIZED TO EXIT SUBROUTINE AFTER
C                        DENSITY IS COMPUTED IF P(X) IS NOT
C                        REQUIRED.
C     ..................................................................
C
```

```
      SUBROUTINE BDTR(X,A,B,P,D,IER,IP,DLBETA)
      DOUBLE PRECISION XX,DLXX,DL1X,AA,BB,G1,G2,G3,G4,DD,PP,
     1XO,FF,FN,XI,SS,CC,RR,DLBETA
C
C        TEST FOR VALID INPUT DATA
C
C     MODIFICATION -- THE FOLLOWING CARD WAS OMITED.
C     IF(A-(.5-1.E-5)) 640,10,10
C
   10 CONTINUE
   20 IF(A-1.E+5) 30,30,640
   30 IF(B-1.E+5) 40,40,640
   40 IF(X) 640,50,50
   50 IF(1.-X) 640,60,60
C
C        COMPUTE LOG(BETA(A,B))
C
   60 AA=DBLE(A)
      BB=DBLE(B)
C
C     MODIFICATION -- THE FOLLOWING CARD WAS ADDED.
C
      IF (IP.EQ.0) GO TO 65
C
      CALL DLGAM(AA,G1,IOK)
      CALL DLGAM(BB,G2,IOK)
      CALL DLGAM(AA+BB,G3,IOK)
      DLBETA=G1+G2-G3
C
C     MODIFICATION -- THE FOLLOWING CARD WAS ADDED.
C
   65 CONTINUE
C
C        TEST FOR X NEAR 0.0 OR 1.0
C
      IF(X-1.E-8) 80,80,70
   70 IF((1.-X)-1.E-8) 130,130,140
   80 P=0.0
      IF(A-1.) 90,100,120
   90 D=1.E+75
      GO TO 660
  100 DD=-DLBETA
      IF(DD+1.68D02)  120,120,110
  110 DD=DEXP(DD)
      D=SNGL(DD)
      GO TO 660
  120 D=0.0
      GO TO 660
  130 P=1.0
      IF(B-1.) 90,100,120
C
C        SET PROGRAM PARAMETERS
C
  140 XX=DBLE(X)
      DLXX=DLOG(XX)
      DL1X=DLOG(1.D0-XX)
      XO=XX/(1.D0-XX)
      ID=0
C
C        COMPUTE ORDINATE
C
      DD=(AA-1.D0)*DLXX+(BB-1.D0)*DL1X-DLBETA
      IF(DD-1.68D02) 150,150,160
  150 IF(DD+1.68D02) 170,170,180
  160 D=1.E75
      GO TO 190
  170 D=0.0
      GO TO 190
```

```
180 DD=DEXP(DD)
    D=SNGL(DD)
C
C   MODIFICATION -- THE FOLLOWING CARD WAS ADDED.
C
    IF (IER.EQ.0) GO TO 670
C
C      A OR B OR BOTH WITHIN 1.E-8 OF 1.0
C
190 IF(ABS(A-1.)-1.E-8)  200,200,210
200 IF(ABS(B-1.)-1.E-8)  220,220,230
210 IF(ABS(B-1.)-1.E-8)  260,260,290
220 P=X
    GO TO 660
230 PP=BB*DL1X
    IF(PP+1.68D02) 240,240,250
240 P=1.0
    GO TO 660
250 PP=DEXP(PP)
    PP=1.D0-PP
    P=SNGL(PP)
    GO TO 600
260 PP=AA*DLXX
    IF(PP+1.68D02) 270,270,280
270 P=0.0
    GO TO 660
280 PP=DEXP(PP)
    P=SNGL(PP)
    GO TO 600
C
C      TEST FOR A OR B GREATER THAN 1000.0
C
290 IF(A-1000.) 300,300,310
300 IF(B-1000.) 330,330,320
310 XX=2.D0*AA/X0
    XS=SNGL(XX)
    AA=2.D0*BB
    DF=SNGL(AA)
    CALL CDTR(XS,DF,P,DUMMY,IER)
    P=1.0-P
    GO TO 670
320 XX=2.D0*BB*X0
    XS=SNGL(XX)
    AA=2.D0*AA
    DF=SNGL(AA)
    CALL CDTR(XS,DF,P,DUMMY,IER)
    GO TO 670
C
C   SELECT PARAMETERS FOR CONTINUED FRACTION COMPUTATION
C
330 IF(X-.5) 340,340,380
340 IF(AA-1.D0) 350,350,360
350 RR=AA+1.D0
    GO TO 370
360 RR=AA
370 DD=DLXX/5.D0
    DD=DEXP(DD)
    DD=(RR-1.D0)-(RR+BB-1.D0)*XX*DD +2.D0
    IF(DD) 420,420,430
380 IF(BB-1.D0) 390,390,400
390 RR=BB+1.D0
    GO TO 410
400 RR=BB
410 DD=DL1X/5.D0
    DD=DEXP(DD)
    DD=(RR-1.D0)-(AA+RR-1.D0)*(1.D0-XX)*DD +2.D0
    IF(DD) 430,430,420
420 ID=1
    FF=DL1X
    DL1X=DLXX
    DLXX=FF
    X0=1.D0/X0
```

```
      FF=AA
      AA=BB
      BB=FF
      G2=G1
C
C         TEST FOR A LESS THAN 1.0
C
  430 FF=0.D0
      IF(AA-1.D0) 440,440,470
  440 CALL DLGAM(AA+1.D0,G4,IOK)
      DD=AA*DLXX+BB*DL1X+G3-G2-G4
      IF(DD+1.68D02) 460,460,450
  450 FF=FF+DEXP(DD)
  460 AA=AA+1.D0
C
C         COMPUTE P USING CONTINUED FRACTION EXPANSION
C
  470 FN=AA+BB-1.D0
      RR=AA-1.D0
      II=80
      XI=DFLOAT(II)
      SS=((BB-XI)*(RR+XI))/((RR+2.D0*XI-1.D0)*(RR+2.D0*XI))
      SS=SS*X0
      DO 480 I=1,79
      II=80-I
      XI=DFLOAT(II)
      DD=(XI*(FN+XI))/((RR+2.D0*XI+1.D0)*(RR+2.D0*XI))
      DD=DD*X0
      CC=((BB-XI)*(RR+XI))/((RR+2.D0*XI-1.D0)*(RR+2.D0*XI))
      CC=CC*X0
      SS=CC/(1.D0+DD/(1.D0-SS))
  480 CONTINUE
      SS=1.D0/(1.D0-SS)
      IF(SS) 650,650,490
  490 CALL DLGAM(AA+BB,G1,IOK)
      CALL DLGAM(AA+1.D0,G4,IOK)
      CC=G1-G2-G4+AA*DLXX+(BB-1.D0)*DL1X
      PP=CC+DLOG(SS)
      IF(PP+1.68D02) 500,500,510
  500 PP=FF
      GO TO 520
  510 PP=DEXP(PP)+FF
  520 IF(ID) 540,540,530
  530 PP=1.D0-PP
  540 P=SNGL(PP)
C
C         SET ERROR INDICATOR
C
      IF(P) 550,570,570
  550 IF(ABS(P)-1.E-7) 560,560,650
  560 P=0.0
      GO TO 660
  570 IF(1.-P) 580,600,600
  580 IF(ABS(1.-P)-1.E-7) 590,590,650
  590 P=1.0
      GO TO 660
  600 IF(P-1.E-8) 610,610,620
  610 P=0.0
      GO TO 660
  620 IF((1.0-P)-1.E-8) 630,630,660
  630 P=1.0
      GO TO 660
  640 IER=-2
      D=-1.E75
      P=-1.E75
      GO TO 670
  650 IER=+2
      P= 1.E75
      GO TO 670
  660 IER=0
  670 RETURN
      END
```

LIST OF REFERENCES

1. Lloyd, D. K., and Lipow, M., Reliability: Management, Methods, and Mathematics, p. 219, Prentice Hall, 1962.

2. Zehna, P. W., Probability Distribution and Statistics, p. 151, Allen and Bacon, 1970.

3. Operations Evaluation Group Research Contribution No. 79, Confidence Limits for System Reliability, by J. Bram, p. 9, 1968.

4. Emshoff, J. R. and Sisson, R. L., Design and Use of Computer Simulation Models, p. 180, Macmillan, 1970.

5. S/360 Scientific Subroutine Package, Version 3, Programmers Manual p. 78-80, 1970.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Chief of Naval Personnel Pers 11b Department of the Navy Washington, D. C. 20370	1
4.	Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
5.	Professor W. M. Woods, Code 53 Wo Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
6.	LCDR Gary Lee Kirk, USN 862 E Avenue Coronado, California 92118	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

A Comparison of Classical and Bayesian Methods for Determining Lower Confidence Limits on System Reliability

4. DESCRIPTIVE NOTES *(Type of report and, inclusive dates)*

Master's Thesis; September 1972

5. AUTHOR(S) *(First name, middle initial, last name)*

Gary Lee Kirk

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1972	49	5

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A series system is simulated to obtain lower confidence limits on system reliability using Bayesian techniques. A comparison between classical and Bayesian methods is made. Random beta variate generators are developed and used in the simulation. The results of the simulation are tabulated for easy comparison of the Bayesian and classical methods. The values of lower confidence limits that are realized using the Bayesian method decrease as the number of components increase. In most cases, as the number of components increase, the Bayesian method appears to yield lower values of lower confidence limits than the classical method.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Bayesian						
Beta Distribution						
Lower Confidence Limits						
Reliability						
System Reliability						